Unreal Bugs Are Real

Presented by
Jeff Paone

Detecting Bugs

- Need a programmatic way to detect runtime errors
- Eliminate trial and error in solving errors
- Do not need to execute code
Existing Work

- FindBugs
  - Developed by University of Maryland
  - Specific to Java
  - Analyzes bytecode (compiled Java files)

- http://findbugs.sourceforge.net/

Domain Specific Languages (DSL)

- Programming language with specific purpose and problem to solve
  - Ex: Mathematica, SQL, UnrealScript
- Uses idioms and terms relevant to domain
- Need to learn new language
UnrealScript

- Specific to *Unreal Tournament*:
- Java based but optimized for simplicity
- Case insensitive
- Does not need to be compiled
- Helped popularize *Unreal* by allowing players to modify the game

---

UnrealScript example

```plaintext
// TriggerLight - A lightsource which can be triggered on or off.
class TriggerLight expands Light;
// Variables.
var() float ChangeTime; // Time light takes to change from on to off.
var ELightType InitialType;
// Engine functions. // Called at start of gameplay.
function BeginPlay() {
    Disable('Tick');
    InitialType = LightType;
    InitialBrightness = LightBrightness;
    if( bInitiallyOn ) { Alpha = 1.0; Direction = 1.0; }
    else { LightType = LT_None; Alpha = 0.0; Direction = -1.0; }
}
```
UnrealScript Bug Detection

- Cannot compile code and run
  - Cannot look at *\.class file like FindBugs
- Should not “trial and error” and detect during gameplay
- Need to parse and debug code

Detecting Bugs
Part 1 - Abstraction

- Read in code
- Classify each line
- Identify variables
- Identify code blocks
- Can find syntax errors at this time
UnrealScript example cont.

// TriggerLight - A lightsource which can be triggered on or off.
class TriggerLight expands Light;

// Variables.
var() float ChangeTime; // Time light takes to change from on to off.
var ELightType InitialType;

// Engine functions. // Called at start of gameplay.
function BeginPlay() {
    Disable( 'Tick' );
    InitialType = LightType;
    InitialBrightness = LightBrightness;
    if ( bInitiallyOn ) { Alpha = 1.0; Direction = 1.0; }
    else { LightType = LT_None; Alpha = 0.0; Direction = -1.0; }
}
Detecting Bugs
Part 2 – Simulate Running

- Check variable values
- Check method calls
- Check loop invariants

- Can find Null Pointers, Incompatible Types, Infinite loops

Results

- Correctly abstracted individual lines and code blocks
  - BUT heavily syntax dependent
  - If ( a < 1 ) { a = b; b++; }
  - If ( a < 1 ) {
    - a = b;
    - b++;
Results cont.

- Diagnosed unseen global variables as undeclared
  - However, these variables come from runtime environment
- Warned if loop did not terminate in $x$ iterations

References

- [http://www.unrealtechnology.com/](http://www.unrealtechnology.com/)
Questions?

- Email me at paonej@colorado.edu
- Post message to the moodle